IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Ashok Singhal, David J. Broniarczyk, George R. Cameron, Jeff A. Price		
Assignee:	3PAR Inc.		
Amended Title:	Communication Link Protocol Optimized for Storage Architecture		
Serial No.:	09/751,649	Filing Date:	December 29, 2000
Examiner:	Steve N. Nguyen	Group Art Unit:	2117
Docket No.:	M-8495 US (3PD-M-8495 US)	Confirmation No.	9244

San Jose, California
August 12, 2008

Mail Stop Appeal Brief - Patents
Under Secretary of Commerce for Intellectual Property and Director of the
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF UNDER 37 CFR §41.37(c)

Dear Sir:

Applicant hereby appeals the rejections of claims 1 to 3 and 10 to 13 in the February 26, 2008 Final Office Action to the Board of Patent Appeals and Interferences.

Please charge the fee of $510.00 for this Appeal Brief as set forth in 37 C.F.R. §41.20(b)(2) to the authorized credit card. Please also charge any amounts underpaid or credit any amounts overpaid to Deposit Account No. 502226.

REAL PARTY IN INTEREST

The real party in interest is the assignee 3PAR, Inc.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

All claims 1 to 3 and 10 to 13 are pending, twice rejected, and appealed.

STATUS OF AMENDMENTS

The Examiner issued the Final Office Action on February 26, 2008. Applicant did not file any amendments subsequent to the rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1

Claim 1 recites a communication link protocol for communicating between a local node and a remote node in an interconnect system. Specification, p. 9, lines 3 to 19. The local node and the remote node are connected by communication links. Id., p. 11, lines 6 to 15. The communication link protocol includes four types of commands. Id., p. 12, lines 3 to 11.

A direct memory access write command provides an inter-node DMA transfer of a data block from the local node to the remote node via one of the communication links. Id., p. 12, lines 12 to 24.

An administrative write command provides a write from the local node to registers in the remote node using one of the communication links. Id., p. 13, line 25 to p. 14, line 8.

A memory copy write command provides, when new data is written to a line of memory at the local node, a copy of the entire line of memory from the local node is written to a corresponding line of memory at the remote node using one of the communication links. Id., p. 10, line 11 to p. 11, line 5; p. 13, lines 15 to 24; p. 28, line 4 to p. 29, line 22. The entire line of memory is sent to

the remote node even when the new data is smaller than the entire line of the memory (i.e., less than the entire line of memory is written). Id.

A built in self test (BIST) command provides a write for testing the functionalities of one of the communication links. Id., p. 14, lines 9 to 16; p. 19, lines 13 to 26.

Claims 2, 3, 10, and 13

Claims 2, 3, 10, and 13 depend from claim 1 and are patentable for at least the same reasons as claim 1.

Claim 11

In addition to the elements recited in claim 1, claim 11 further recite that the inter-node DMA transfer includes computing parity over multiple data blocks from a local memory of the local node and writing the parity to a remote memory of the remote node in the very same operation as the transfer of the data block. Specification, p. 12, line 25 to p. 13, line 14; p. 19, lines 3 to 12.

Claim 12

In addition to the elements recited in claim 1, claim 12 further recite that the memory copy write command includes (1) reading existing data from the line of memory in a local memory of local node, (2) merging the new data with the existing data so the new data replaces at least some of the existing data, (3) writing the merged data to the line of memory in the local memory of the local node, (4) transferring the merged data using one of the communication links to the remote node, and (5) writing the merged data to the corresponding line of memory in a remote memory of the remote node. Specification, p. 13, lines 15 to 24; p. 28, line 4 to p. 29, line 22.

Claim 13

In addition to the elements recited in claims 1 and 12, claim 13 recites that the merged data is written to the corresponding line of memory in the remote memory of the remote node using a same address offset of the line of memory at the local memory of the local node. Specification, p. 26, Table 4 ("[f]or memory copy writes, the ADDR field contains an offset from the base address of the sending node's send range, which is used as an offset from the base of the receiving node's receive range for the respective communication link").

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The Examiner rejected claims 1 to 3, 10, and 12 under 35 U.S.C. § 103(a) as being unpatentable over the combination of U.S. Patent No. 6,049,889 ("Steely Jr. et al."), U.S. Patent No. 5,858,556 ("Grivna"), and U.S. Patent No. 5,887,134 ("Ebrahim"). The Examiner rejected claim 13 under 35 U.S.C. § 103(a) as being unpatentable over the combination of Steely, Jr. et al., Grivna, Ebrahim, and U.S. Patent No. 6,038,677 ("Lawlor et al."). The Examiner rejected claim 11 under 35 U.S.C. § 103(a) as being unpatentable over the combination of Steely, Jr. et al., Grivna, Ebrahim, and U.S. Patent No. 5,914,970 ("Gunsaulus et al.").

ARGUMENTS

Claim 1

The Examiner cited Steely Jr. et al. for disclosing a memory copy write command that copies an entire line of memory from a local node to a remote node even when the line is only partially written with new data. The Examiner stated:

> The Examiner notes that Steely teaches a reflected memory write between nodes in col. 6, lines 46-47 and col. 7, lines 13-15. In a reflected memory write, written data is reflected to the other nodes. Steely discloses that connection granularity between nodes is at the page level (col. 4, lines 54-57), which means that the smallest unit of data transmitted between nodes is 8K bytes in the example in col. 4, lines 54-57. Therefore, data smaller than 8k bytes that is written to memory will still be transmitted in a page of 8k bytes by using pad bits or any other method commonly known in the art. This makes sense because memories operate on units of data- in this particular case 8k bytes.

February 26, 2008 Final Office Action, pp. 2 and 3. Applicant respectfully traverses.

Steely Jr. et al. does not disclose a memory copy write command that mirrors data in a line of memory at a local node to a corresponding line of memory at a remote node after the line of memory at the local node has been written. In other words, the data is written into the memory of the local node and a copy of the data is written into the memory of the remote node for safe keeping. This prevents data written to a node (e.g., by a host) from being lost if the node fails as the data is mirrored to another node for backup purposes.

Steely Jr. et al. does not disclose that data is written to a local memory of a local node and then mirrored to a remote memory at a remote node. Instead, Steely Jr. et al. discloses a reflected

memory write where a write to a specified address space at the local node is mapped to an address at the remote node and then sent to that address at the remote node. Steely Jr. et al. provides an example of the reflected memory write as follows.

> For example, referring now to FIG. 5, an example write of 32B across the data link 20 from Node 1 to node 2 is shown to include the following steps. First, at step 60, the CPU 22 performs a sequence of 4 Store Quad instructions to an aligned 32 byte address in PCI address space, where each Store Quad instruction has the effect of storing 8 bytes of information. At step 62, the 4, 8 byte stores are converted by the CPU 22 into one aligned 32 byte store command. At step 64, the I/O interface 28 translates the 32 byte store command into a 32-byte PCI write to the corresponding MC address portion of PCI memory space. At step 66, the PCI to MC adapter 34 checks the address of the write command to see if it is to MC address space. If it is, at step 68 the PCI to MC adapter 34 accepts the write, converts it into a 32 byte MC write to the corresponding network address and transmits the request over the data link 20. To convert a PCI address to an MC address, bits <31:27> of the original address are replaced with the contents of the MC base address register 53. The address is then extended to a full 40 bits by assigning zeros to bits <39:32>. At step 70, the PCI-MC adapter at the receiving node accepts the MC write and converts it to a 32 byte PCI write to the corresponding MC page. At step 72, the I/O interface at the receiving node accepts the write and converts it to a 32 byte write to local memory space with an address defined by a corresponding DMA scatter/gather map 57 (FIG. 4).

Steely Jr. et al., col. 5, line 63 to col. 6, line 21 (emphasis added). As the above paragraph states, CPU 22 performs a write to an address in the PCI address space, and PCI to MC adapter 34 determines if the address is mapped to an address in the MC address space. If so, then PCI to MC adapter 34 translates the PCI address to a MC address, and sends the data with the MC address over data link 20 to the other node. Thus, Steely Jr. et al. does not ever disclose that the data is written into the memory of the local node. Instead, the address is translated and the data is sent to the memory of the remote node.

Furthermore, Steely Jr. et al. does not disclose or suggest that an entire line of memory at a local node is mirrored to a corresponding line of memory at a remote node when less than the entire line at the local node is written. Steely Jr. et al. only discloses that PCI address space 47 at a node is divided into pages and each page is connected (i.e., assigned) to one or more nodes in the system.

> Thus, connection granularity between nodes in the network is at the page level. Certain nodes in the network will receive data when the CPU writes to one of the N pages of MC address space. The determination of which nodes are

> mapped to which network addresses, i.e. the mapped connection, are determined at some point prior to when the nodes require data transfer.

Steely Jr. et al. col. 4, lines 56 to 64 (emphasis added). Connection granularity simply refers to the correspondence between pages and nodes to which they are assigned. Referring to the above quoted paragraph of Steely Jr. et al., CPU 22 at the local node writes 32 bytes to a PCI address space, and PCI to MC adapter 34 at the local node transfers the 32 bytes to the PCI to MC adapter at the remote node. Steely Jr. et al., col. 5, line 63 to col. 6, line 21. The above quoted paragraph and the remainder of Steely Jr. et al. simply makes no mention that an entire page (i.e., 8 K bytes) is transferred between the nodes when less than the entire page is written.

Grivna and Ebrahim do not cure the deficiencies of Steely Jr. et al. For the above reasons, claim 1 is patentable over the combination of Steely Jr. et al., Grivna, and Ebrahim.

Claims 2, 3, and 10

Claims 2, 3, and 10 depend from claim 1 and are patentable for at least the same reasons as claim 1.

Claim 11

The Examiner found that Steely et al., Grivna, and Ebrahim do not disclose computing parity over multiple blocks of data from a local memory at a local node and writing the parity to a remote memory of a remote node. The Examiner then cited U.S. Patent No. 5,914,970 ("Gunsaulus et al."). for disclosing "a single operation of computing parity and writing the parity in the destination parity memory in col. 4, lines 29-33." February 26, 2008 Final Office Action, p. 3. Furthermore, "[b]rought into the context of Steely and Ebrahim, it would have been obvious to one skilled to write the parity directly to the remote node since that is the destination node." Id. Applicant respectfully traverses.

Gunsaulus et al. discloses that, when a processor sub system 12 writes data to a word 34 (or a byte 36 of word 34) in a memory 14, it also calculates parity for the data and stores the parity in memory 14. Thus, the parity is calculated in response to data being written in memory 14. On the other hand, claim 11 recites a DMA write command that causes parity to be calculated at a local node and then written to a remote node. In other words, the steps of calculating parity and writing the parity from a local node to a remote node have been combined under a single command. Thus,

unlike Gunsaulus et al., the parity is not generated in response to data being written (whether to a local memory or a remote memory) but the parity itself is the data being written.

For all of the above reasons, claim 11 is patentable over the cited references.

Claim 12

The Examiner cited Steely Jr. et al., col. 4, lines 54 to 57 and col. 7, lines 13 to 15, for teaching the details of the memory copy write command as recited in claim 12. Applicant respectfully traverses.

As discussed above, col. 4, lines 54 to 57 and col. 7, lines 13 to 15 of Steely Jr. et al. disclose that PCI address space at a node is divided into pages and each page is connected (i.e., assigned) to one or more nodes in the system. As discussed above, Steely Jr. et al. does not disclose an entire page at a node is mirrored to another node when less than the entire page is written, let alone the specific steps recited in claim 12.

For all of the above reasons, claim 12 is patentable over the cited references.

Claim 13

The Examiner cited Lawlor et al. to show that it was well known to identically replicate data of a local node at a remote node so that one skilled in the art would modify Steely Jr. et al. so that the data would be written with the same address offsets. Applicant respectfully traverses.

Steely, Jr. et al. discloses that address translation is used to write data in local memory space of a transmitting node to a network address space, and then from the network address space back to a local memory space of a receiving node.

> For example, writes to the shared portion of memory address space 43 are translated by map 43a to an address in network address space. The network address is translated by map 44a in node 14 to an address of the node memory of node 14. Accordingly, node 12 communicates with node 14 via writes its own MC address space. Similarly, writes to the shared portion of memory address space 34 by node 14 are translated by map 44a to an address in network address space 33. The network address is translated by map 43a of node 12 into a node memory address for node 12. Such an arrangement allows for communication between the CPU or external I/O devices of node 12 and the CPU or external I/O device of node 14 by providing memory-mapped connections which are established between the nodes.

Steely Jr. et al., col. 4, lines 16 to 29 (emphasis added). By eliminating address translation, the Examiner has changed the principle of operation of Steely Jr. et al., which is prohibited by MPEP § 2143.01. Therefore, there is no motivation to modify Steely Jr. et al. with Lawlor et al. as suggested by the Examiner.

CONCLUSION

Applicant respectfully submits the Examiner has failed to show that the cited references disclose all the recited elements of claims 1 to 3 and 10 to 13. Accordingly, Applicant requests the rejections of claims 1 to 3 and 10 to 13 to be reversed.

Respectfully submitted,

/David C Hsia/

David C. Hsia
Attorney for Applicant(s)
Reg. No. 46,235

Patent Law Group LLP
2635 North First St., Ste. 223
San Jose, California 95134
408-382-0480x206

CLAIMS APPENDIX

Claim 1: A communication link protocol for communicating between a local node and a remote node of an interconnect system via communication links, the communication link protocol comprising:

a direct memory access (DMA) write command for performing an inter-node DMA transfer of a block of data directly from the local node to the remote node via one of the communication links;

an administrative write command for writing data from the local node to registers in the remote node via one of the communication links for administrative purposes;

a memory copy write command for copying an entire line of memory from the local node to a corresponding line of memory at the remote node via one of the communication links after a new data is written into the line of memory at the local node even when the new data is smaller than the line of memory at the local node; and

a built in self test (BIST) command for testing the functionality of one of the communication links.

Claim 2: The communication link protocol of Claim 1 wherein each command is conveyed between the local node and the remote node in the form of a respective command packet.

Claim 3: The communication link protocol of Claim 2 wherein each respective command packet carries information for at least one command flag.

Claim 10: The communication link protocol of Claim 1, wherein said performing an inter-node DMA transfer of a block of data directly from the local node to the remote node comprises copying the block of data from a local memory of the local node to a remote memory of the remote node.

Claim 11: The communication link protocol of Claim 1, wherein said performing an inter-node DMA transfer of a block of data directly from the local node to the remote node comprises computing parity over multiple blocks of data from a local memory of the local node and writing the parity to a remote memory of the remote node in a single operation.

Claim 12: The communication link protocol of Claim 1, wherein the memory copy write command comprises:

reading existing data from the line of memory in a local memory of the local node;

merging the new data with the existing data so the new data replaces at least some existing data while other existing data remains;

writing merged data to the line of memory in the local memory of the local node;

transferring the merged data via one of the communication links to the remote node; and

writing the merged data to the corresponding line of memory in a remote memory of the remote node.

Claim 13: The communication link protocol of Claim 12, wherein said writing the merged data to the corresponding line of memory in a remote memory of the remote node comprises writing the remote node using a same address offset of the line of memory at the local memory of the local node.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None.